APPLICATIONS OF LINEAR ALGEBRA

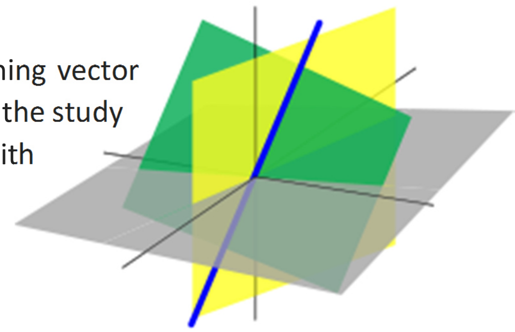
IN SOFTWARE ENGINEERING

FEBRUARY 22, 2016

SUBMITTED BY: M.REHAN ASGHAR
BSSE (4) 15126

Applications of Linear Algebra

Definition:



Linear algebra is the branch of mathematics concerning vector spaces and **linear** mappings between such spaces. It includes the study of lines, planes, and subspaces, but is also concerned with properties common to all vector spaces.

Applications:

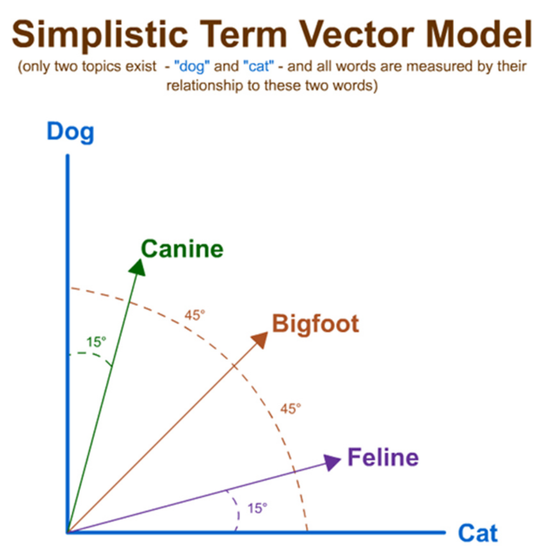
Linear algebra is used in many fields of mathematics, natural sciences, **computer science**, and social science. Below are just some examples of applications of linear algebra in software engineering.

i. Internet search
ii. Graph analysis
iii. Machine learning
iv. Graphics
v. Bioinformatics
vi. Scientific computing
vii. Data mining
viii. Computer vision
ix. Speech recognition
x. Compilers
xi. Parallel computing

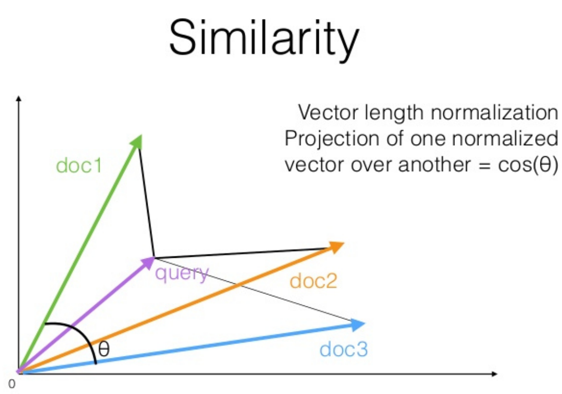
We will discuss few of these examples in detail.

1. Internet Search:

We will discuss about **Vector Space Model Search Engines**. Another information retrieval technique uses the vector space model (Salton, 1971), developed by Gerard Salton in the early 1960's, to sidestep some of the information retrieval problems of the Boolean model. Vector space models transform textual data into numeric vectors and matrices, then employ matrix analysis techniques to discern key features and connections in the document collection.



Some advanced vector space models address the common text analysis problems of synonymy and polysemy. Models such as Latent Semantic Indexing (LSI) (Dumais, 1991) can access the hidden semantic structure in a document collection. For example, an LSI engine processing the query car will return documents whose keywords are related semantically (in meaning), e.g., automobile. This ability to reveal hidden semantic meanings makes vector space models such as LSI very powerful information retrieval tools.



Two additional advantages of the vector space model are relevance scoring and relevance feedback. The model allows documents to match a query partially by assigning each document a number between 0 and 1, which can be interpreted as the likelihood of relevance to the query. The group of retrieved documents can then be sorted by degree of relevancy, a luxury not possible with the Boolean model. Thus, vector space models return documents in an ordered list, with the first document judged to be most relevant to the user's query. Some vector space search engines report the relevance score as a relevancy percentage. For example, a 97% next to a document means that document is judged as 97% relevant to the user's query.

Example. The Federal Communications Commission's search engine, which is powered by Inktomi, was known at one time to use a vector space model. Enter a query such as taxes and notice the relevancy score reported on the right side.

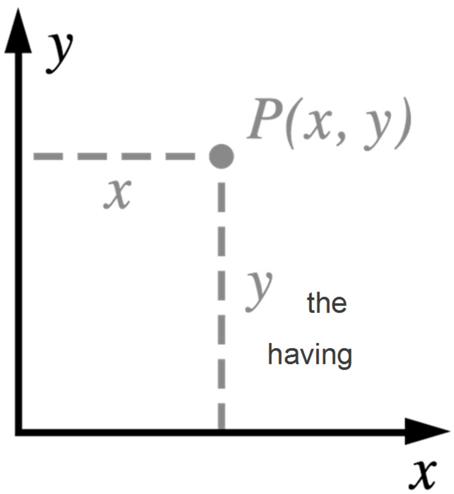
Relevance feedback is an information retrieval tuning technique that is a natural addition to the vector space model. It allows the user to select a subset of the retrieved documents that are useful and then resubmit with this additional relevance feedback information to obtain a revised set of generally more useful retrieved documents.

The drawbacks of vector space models are their computational expense and poor scalability. At query time, distance measures (also known as similarity measures) must be computed between each document and the query, and advanced models such as LSI require an expensive singular value decomposition of a large matrix that numerically represents the entire document collection. As the collection grows, the expense of this matrix decomposition becomes prohibitive.

2. Linear Algebra for Graphics Programming

In order to make the move to three-dimensional space, we need to introduce a few new mathematical concepts. We need to know how to represent points in 3D space, how to move points between different coordinate frames, and how to remove the third dimension when projecting points onto the screen.

The Cartesian plane



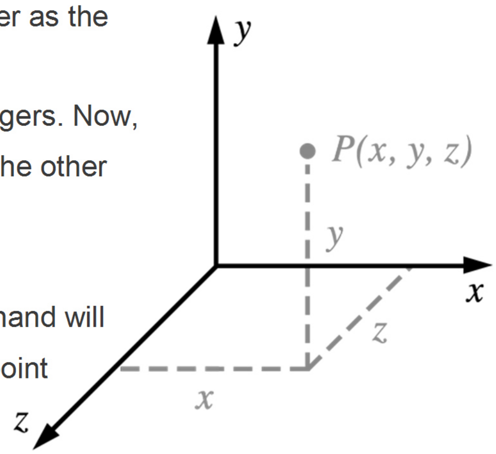
The Cartesian plane has two perpendicular axes (commonly labeled). When we want to speak of things that are perpendicular, we'll call them *orthogonal*. Points are identified by specifying their extent along each axis. For example, the point is 3 units to the right along the x axis and 5 units up along the y axis, relative to the origin. The origin is identified as the point having coordinates (0, 0).

We turn the Cartesian plane into a 3D coordinate space by adding another orthogonal axis,

Handedness

Handedness refers to the orientation of the z-axis in a given 3D space. If the z-axis conforms to the so-called right-hand rule, then the space is said to be *right-handed*. Alternatively, if the z-axis points in the other direction, the space is left-handed.

The term "handedness" comes from a mnemonic for remembering which way the third axis points. Identify your thumb as the x axis and your pointer finger as the



y axis. Hold up either hand and make an 'L' with these two fingers. Now, when you extend your middle finger perpendicular to both of the other fingers, it indicates the

direction of the positive z axis: the middle finger of your right hand will point toward you, and the middle finger of your left hand will point away.

The choice is arbitrary, but I choose to work in right-handed spaces when possible. However, we will give Metal our vertices in a 3D space called *clip space*, which is left-handed. As long as we make the switch from right- to left-handed at the correct place in the rendering process, everything works out okay.

Introduction to Transformations

A geometric transformation is a function that maps a point to another point. The most common transformations in computer graphics are translation, rotation, and scaling. In three dimensions, rotation and scaling can be represented as a multiplication of a 3×3 matrix by a 3D point. Unfortunately, translation cannot be represented in this way, but there is a formulation we'll see below that nevertheless allows us to capture all the transformations we wish to perform using matrix multiplication.

First, we'll consider the family of transformations known as *linear transformations*.

Linear Transformations

A linear transformation must obey these two properties. The first condition means that scaling the input before the transformation is the same as scaling the output after the transformation. The second condition means that the transformation of sums is equal to the sum of the transformed inputs.